**Introduction**

Toutes les grandes industries implémentent Apache Hadoop en tant que cadre standard pour le traitement et le stockage de données volumineuses. Hadoop est conçu pour être déployé sur un réseau de centaines voire de milliers de [serveurs](https://phoenixnap.com/servers/dedicated). Toutes ces machines fonctionnent ensemble pour traiter le volume massif et la variété des ensembles de données entrants.

Le déploiement des services Hadoop sur un seul nœud est un excellent moyen de se familiariser avec les commandes et les concepts de base de Hadoop.

**Installer OpenJDK sur Ubuntu**

Le framework Hadoop est écrit en Java et ses services nécessitent un environnement d'exécution Java (JRE) et un kit de développement Java (JDK) compatibles. Utilisez la commande suivante pour mettre à jour votre système avant de lancer une nouvelle installation :

sudo apt update

Pour le moment, **Apache Hadoop 3.x prend entièrement en charge Java 8.** Le package OpenJDK 8 dans Ubuntu contient à la fois l'environnement d'exécution et le kit de développement.

Tapez la commande suivante dans votre terminal pour installer OpenJDK 8 :

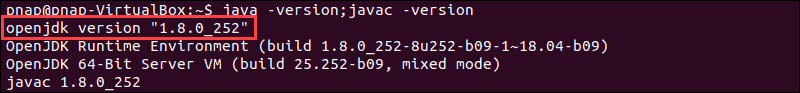
sudo apt install openjdk-8-jdk -y

La version OpenJDK ou Oracle Java peut affecter la façon dont les éléments d'un écosystème Hadoop interagissent.

Une fois le processus d'installation terminé, vérifiez la version actuelle de Java :

java -version; javac -version

La sortie vous informe de l'édition Java utilisée.



**Configurer un utilisateur différent de root pour l'environnement Hadoop**

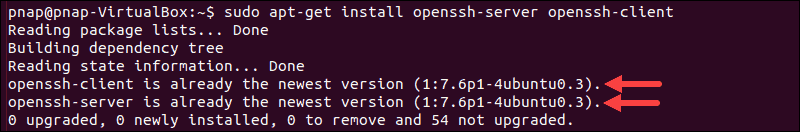
Il est conseillé de créer un utilisateur différent de root, spécifiquement pour l'environnement Hadoop. Un utilisateur distinct améliore la sécurité et vous aide à gérer votre cluster plus efficacement. Pour assurer le bon fonctionnement des services Hadoop, l'utilisateur doit avoir la possibilité d'établir une connexion SSH sans mot de passe avec le localhost.

**Installer OpenSSH sur Ubuntu**

Installez le serveur et le client OpenSSH à l'aide de la commande suivante :

sudo apt install openssh-server openssh-client -y

Dans l'exemple ci-dessous, la sortie confirme que la dernière version est déjà installée.



Si vous avez installé OpenSSH pour la première fois, profitez de cette occasion pour mettre en œuvre ces recommandations de sécurité SSH vitales.

**Créer un utilisateur Hadoop**

Utilisez la commande **adduser** pour créer un nouvel utilisateur Hadoop :

sudo adduser hadoop

Le nom d'utilisateur, dans cet exemple, est **hdoop**. Vous êtes libre d'utiliser le nom d'utilisateur et le mot de passe que vous jugez appropriés. Basculez vers l'utilisateur nouvellement créé et saisissez le mot de passe correspondant :

su - hadoop

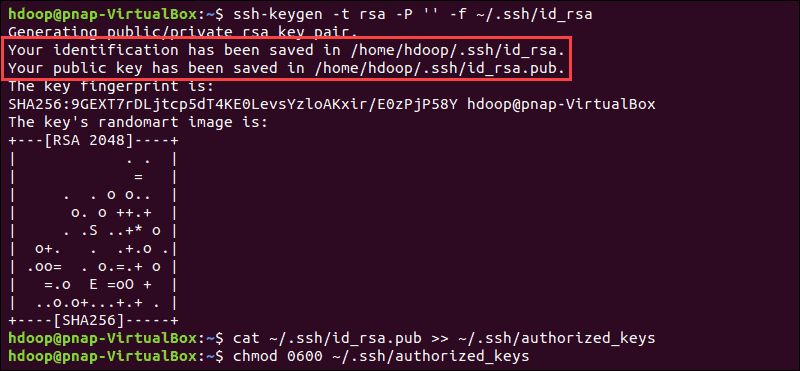
L'utilisateur doit maintenant pouvoir se connecter en SSH à l' hôte local sans être invité à entrer un mot de passe.

**Activer SSH sans mot de passe pour l'utilisateur Hadoop**

Générez une paire de clés SSH et définissez l'emplacement dans lequel elle doit être stockée :

ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa

Le système procède à la génération et à l'enregistrement de la paire de clés SSH.



Utilisez la commande **cat** pour stocker la clé publique en tant que **authorized\_keys**dans le répertoire *ssh* :

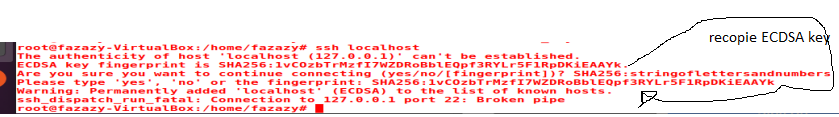
cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

Définissez les autorisations de votre utilisateur avec la commande **chmod** :

chmod 0600 ~/.ssh/authorized\_keys

Le nouvel utilisateur peut désormais se connecter en SSH sans avoir à saisir un mot de passe à chaque fois. Vérifiez que tout est correctement configuré en utilisant l’utilisateur **hdoop** pour SSH vers localhost :

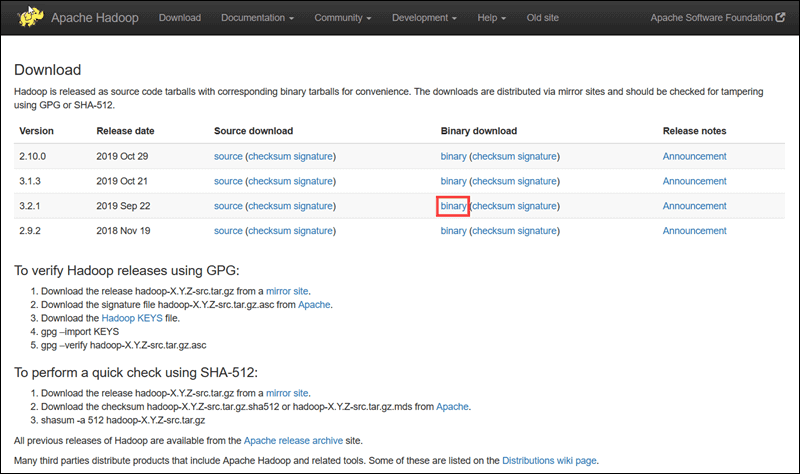
ssh localhost



Après une première invite, l'utilisateur Hadoop est désormais en mesure d'établir une connexion SSH à l’hôte local de manière transparente.

**Télécharger et installer Hadoop sur Ubuntu**

Visitez la [page officielle du projet Apache Hadoop](https://hadoop.apache.org/releases.html) et sélectionnez la version de Hadoop que vous souhaitez implémenter.



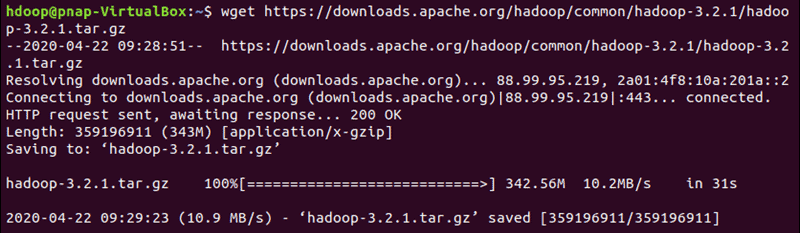
Les étapes décrites dans ce didacticiel utilisent le téléchargement binaire pour **Hadoop version 3.2.1**.

Sélectionnez l’option préférée et vous obtenez un lien miroir qui vous permet de télécharger le **package tar Hadoop**.



Utilisez le lien miroir fourni et téléchargez le package Hadoop avec la commande **wget** :

wget https://downloads.apache.org/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz



Une fois le téléchargement terminé, extrayez les fichiers pour lancer l'installation de Hadoop :

tar xzf hadoop-3.2.1.tar.gz

Les fichiers binaires Hadoop se trouvent désormais dans le répertoire *hadoop-3.2.1* .

**Déploiement Hadoop à nœud unique (mode pseudo-distribué)**

Hadoop excelle lorsqu'il est déployé en **mode entièrement distribué** sur un grand cluster de serveurs en réseau. Toutefois, si vous débutez avec Hadoop et que vous souhaitez explorer les commandes de base ou tester des applications, vous pouvez configurer Hadoop sur un seul nœud.

Cette configuration, également appelée **mode pseudo-distribué,** permet à chaque démon Hadoop de s'exécuter comme un processus Java unique. Un environnement Hadoop est configuré en modifiant un ensemble de fichiers de configuration :

* + - bashrc
    - hadoop-env.sh
    - core-site.xml
    - hdfs-site.xml
    - mapred-site-xml
    - yarn-site.xml

**Configurer les variables d'environnement Hadoop ( bashrc )**

Modifiez le *fichier de configuration du shell bashrc* à l'aide d'un éditeur de texte de votre choix (nano ):

sudo nano . bashrc

Définissez les variables d'environnement Hadoop en ajoutant le contenu suivant à la fin du fichier :

#Hadoop Related Options

export HADOOP\_HOME=/home/hadoop/hadoop-3.2.1

export HADOOP\_INSTALL=$HADOOP\_HOME

export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_HOME=$HADOOP\_HOME

export HADOOP\_HDFS\_HOME=$HADOOP\_HOME

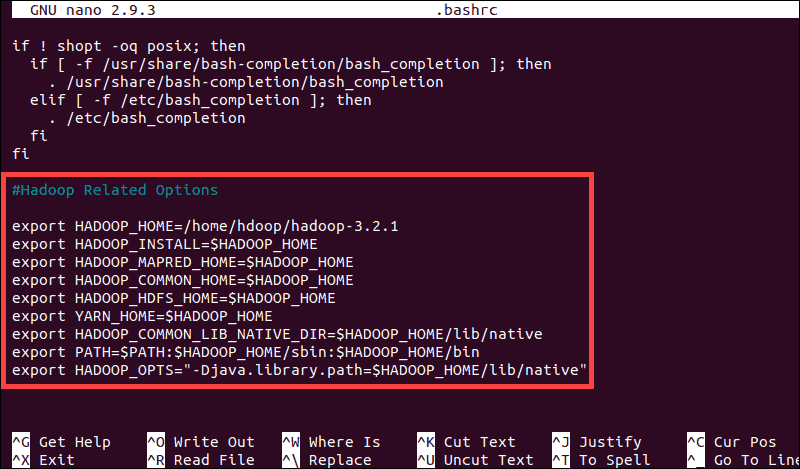
export YARN\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native

export PATH=$PATH:$HADOOP\_HOME/sbin:$HADOOP\_HOME/bin

export HADOOP\_OPTS= "-Djava.library.path=$HADOOP\_HOME /lib/nativ"

Une fois que vous avez ajouté les variables, enregistrez et quittez le fichier .*bashrc*



Il est essentiel d'appliquer les modifications à l'environnement d'exécution actuel à l'aide de la commande suivante :

source ~/.bashrc

**Modifier le fichier hadoop-env.sh**

Le fichier *hadoop-env.sh* sert de fichier maître pour configurer YARN, HDFS, MapReduce et les paramètres de projet liés à Hadoop.

Lors de la configuration d'un (**single node Hadoop cluster)**, vous devez définir l'implémentation Java à utiliser. Utilisez la variable **$HADOOP\_HOME** créée précédemment pour accéder au fichier *hadoop-env.sh* :

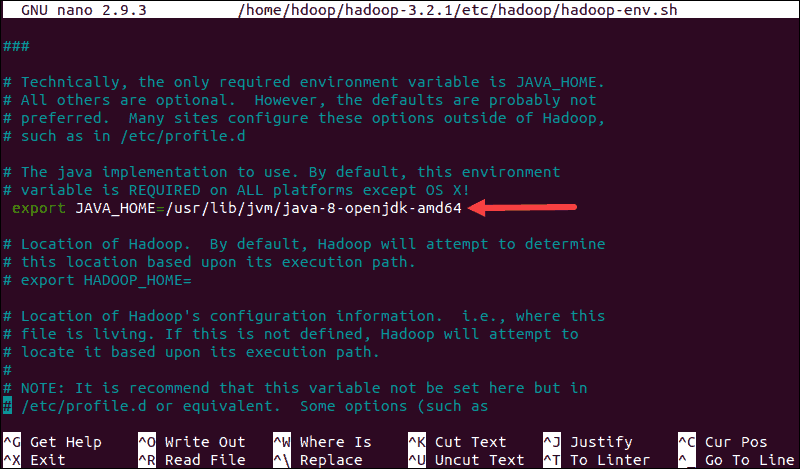
sudo nano $HADOOP\_HOME/etc/hadoop/hadoop-env.sh

Décommentez la variable **$JAVA\_HOME** (supprimez le signe **#**) et ajoutez le chemin d'accès complet à l'installation d' OpenJDK sur votre système. Si vous avez installé la même version que celle présentée, ajoutez la ligne suivante :

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

NB : supprimer # avant #export

Le chemin doit correspondre à l'emplacement de l'installation de Java sur votre système.



Si vous avez besoin d'aide pour localiser le chemin Java correct, exécutez la commande suivante dans la fenêtre de votre terminal :

which javac

La sortie résultante fournit le chemin d'accès au répertoire binaire Java.

Java binary directory location with which command.

Utilisez le chemin fourni pour trouver le répertoire OpenJDK avec la commande suivante :

readlink -f /usr/bin/javac

La section du chemin juste avant le répertoire */bin/ javac* doit être affectée à la variable **$JAVA\_HOME .**

The location of the openjdk binary directory.

**Modifier le fichier core-site.xml**

Le fichier *core-site.xml* définit les propriétés principales HDFS et Hadoop.

Pour configurer Hadoop en mode pseudo-distribué, vous devez **spécifier l'URL** de votre NameNode et le répertoire temporaire utilisé par Hadoop pour le mappage et le processus de réduction.

Ouvrez le fichier *core-site.xml* dans un éditeur de texte :

sudo nano $HADOOP\_HOME/etc/hadoop/core-site.xml

Ajoutez la configuration suivante pour remplacer les valeurs par défaut du répertoire temporaire et ajoutez votre URL HDFS pour remplacer le paramètre de système de fichiers local par défaut :

<configuration>

<property>

<name>hadoop.tmp.dir</name>

<value>/home/hdoop/tmpdata</value>

</property>

<property>

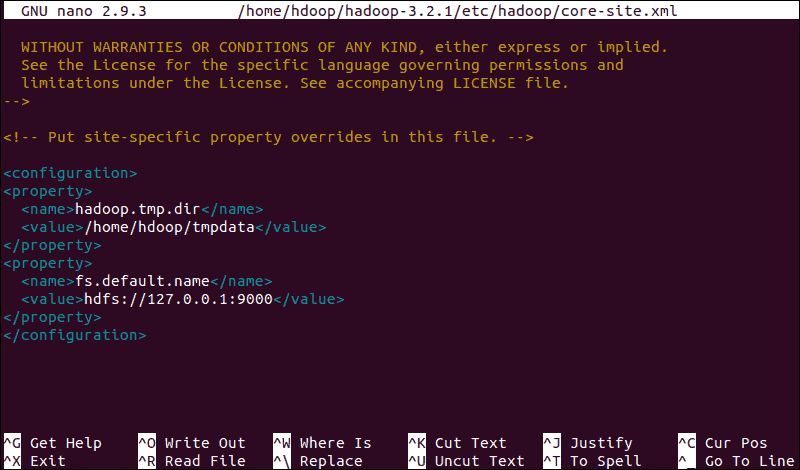
<name>fs.default.name</name>

<value>hdfs://127.0.0.1:9000</value>

</property>

</configuration>

Cet exemple utilise des valeurs spécifiques au système local. Vous devez utiliser des valeurs qui correspondent aux exigences de votre système. Les données doivent être cohérentes tout au long du processus de configuration.



N'oubliez pas de créer un répertoire Linux à l'emplacement que vous avez spécifié pour vos données temporaires.

**Modifier le fichier hdfs-site.xml**

Les propriétés du fichier *hdfs-site.xml* régissent l'emplacement de stockage des métadonnées de nœud, du fichier fsimage et du fichier journal **editlog**. Configurez le fichier en définissant les répertoires de stockage **NameNode** et **DataNode** .

De plus, la valeur par défaut **dfs.replication** de **3** doit être changée en **1** pour correspondre à la configuration de nœud unique.

Utilisez la commande suivante pour ouvrir le fichier *hdfs-site.xml* afin de le modifier :

sudo nano $HADOOP\_HOME/etc/hadoop/hdfs-site.xml

Ajoutez la configuration suivante au fichier et, si nécessaire, ajustez les répertoires NameNode et DataNode à vos emplacements personnalisés :

<configuration>

<property>

<name>dfs.data.dir</name>

<value>/home/hdoop/dfsdata/namenode</value>

</property>

<property>

<name>dfs.data.dir</name>

<value>/home/hdoop/dfsdata/datanode</value>

</property>

<property>

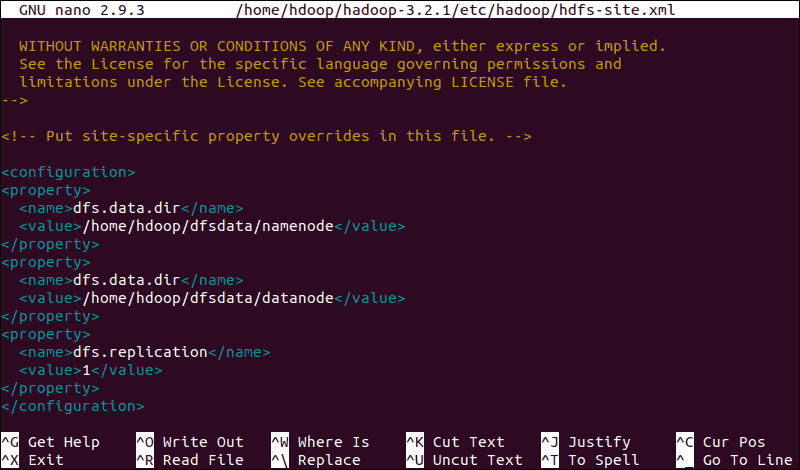
<name>dfs.replication</name>

<value>1</value>

</property>

</configuration>

Si nécessaire, créez les répertoires spécifiques que vous avez définis pour la valeur **dfs.data.dir**.

****

**Modifier le fichier mapred-site.xml**

Utilisez la commande suivante pour accéder au fichier mapred-site.xml et définir les valeurs MapReduce :

sudo nano $HADOOP\_HOME/etc/hadoop/mapred-site.xml

Ajoutez la configuration suivante pour modifier la valeur par défaut du nom de framework MapReduce en **yarn** :

<configuration>

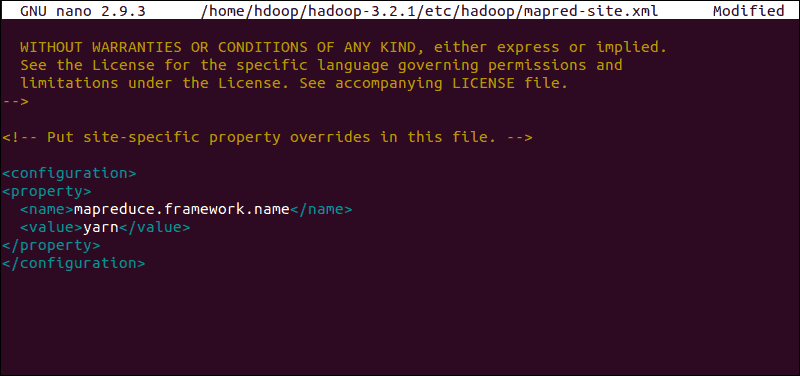
<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>



**Modifier le fichier Yarn-site.xml**

Le fichier *yarn-site.xml* est utilisé pour définir les paramètres relatifs à **YARN**. Il contient des configurations pour le gestionnaire de **nœuds, le gestionnaire de ressources, les conteneurs** et **le maître d’application.**

Ouvrez le fichier *yarn-site.xml* dans un éditeur de texte :

sudo nano $HADOOP\_HOME/etc/hadoop/yarn-site.xml

Ajoutez la configuration suivante au fichier :

<configuration>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

<property>

<name>yarn.resourcemanager.hostname</name>

<value>127.0.0.1</value>

</property>

<property>

<name>yarn.acl.enable</name>

<value>0</value>

</property>

<property>

<name>yarn.nodemanager.env-whitelist</name>

<value>JAVA\_HOME,HADOOP\_COMMON\_HOME,HADOOP\_HDFS\_HOME,HADOOP\_CONF\_DIR,CLASSPATH\_PERPEND\_DISTCACHE,HADOOP\_YARN\_HOME,HADOOP\_MAPRED\_HOME</value>

</property>

</configuration>



**Formater le nœud de nom HDFS**

Il est important de **formater le NameNode** avant de démarrer les services Hadoop pour la première fois :

hdfs namenode -format

La notification d'arrêt signifie la fin du processus de formatage du NameNode.

Formater l'écosystème hadoop

* arrêter le hadoop $HADOOP\_HOME ./ sbin /stop-all.sh
* Puis formatez l' écosystème hadoop

hadoop namenode -format

hadoop datanode -format

hdf namenode -format

hdf datanode -format

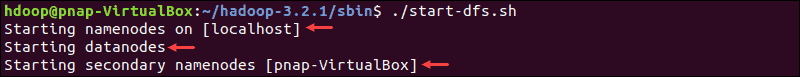
* Puis recommencez en utilisant ./ sbin /start-all.sh

**Démarrer le cluster Hadoop**

Accédez au répertoire *hadoop-3.2.1/sbin* et exécutez les commandes suivantes pour démarrer le NameNode et le DataNode :

Le système prend quelques instants pour initier les nœuds nécessaires.

./start-dfs.sh

  
Une fois que le namenode , les datanodes et le namenode secondaire sont opérationnels, démarrez la ressource YARN et les gestionnaires de nœuds en tapant :

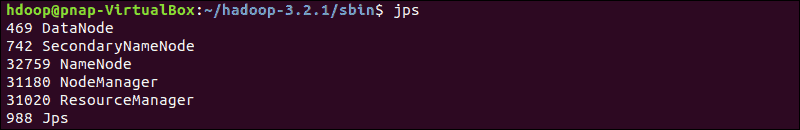
./start-yarn.sh

Comme avec la commande précédente, la sortie vous informe que les processus démarrent.

The system initiates the resource and node manager daemons.  
Tapez cette commande simple pour vérifier si tous les démons sont actifs et s'exécutent en tant que processus Java :

jps

Si tout fonctionne comme prévu, la liste résultante des processus Java en cours d'exécution contient tous les démons HDFS et YARN.

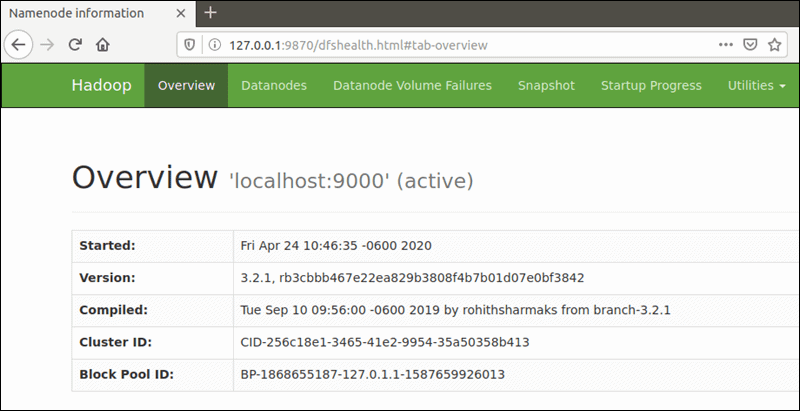


**Accéder à l'interface utilisateur Hadoop depuis le navigateur**

Utilisez votre navigateur préféré et accédez à l' URL ou à l'adresse IP de votre hôte local . Le numéro de port par défaut **9870** vous donne accès à l'interface utilisateur Hadoop NameNode :

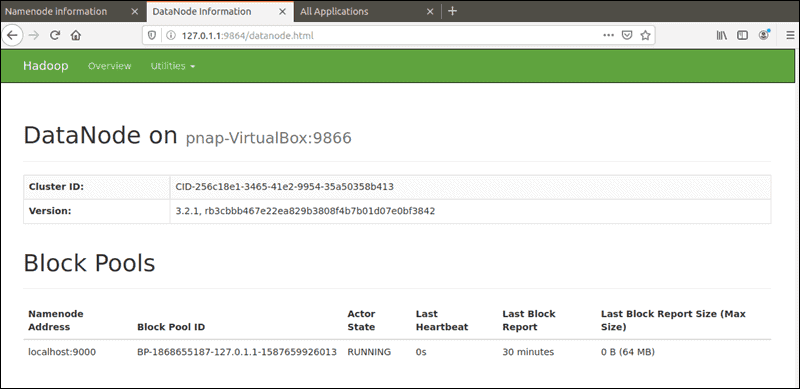
http://localhost:9870

L'interface utilisateur NameNode fournit une vue d'ensemble complète de l'ensemble du cluster.



Le port par défaut **9864** est utilisé pour accéder à des DataNodes individuels directement depuis votre navigateur :

http://localhost:9864



Le gestionnaire de ressources YARN est accessible sur le port **8088** :

http://localhost:8088

Le gestionnaire de ressources est un outil inestimable qui vous permet de surveiller tous les processus en cours d'exécution dans votre cluster Hadoop.

